REMARKS

Attached hereto are a request for a three-month extension of time and the appropriate fee.

Applicants have amended Claims 1-3, 13, 17, 20, 24, 27, 28, 31, 35, 44, 47, 51, and 54, and have added new Claims 56-57. Currently, Claims 1-57 are pending in the application.

The Office Action objected to the drawings, stating that Figures 1-8 should be designated with a legend such as "Prior Art." Applicants have complied with the Examiner's suggestion, and have submitted a Letter to Official Draftsperson adding "Prior Art" to each of Figures 1-12.

The Office Action rejected Claims 3 and 13 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Specifically, the Office Action stated that "judging means for judging" is vague, and suggested that "judging" be replaced with "determining." In response, Applicants have changed occurrences of "judging" to "determining" in Claims 3 and 13, and in other claims as well.

Regarding the Office Action's objection to use of the word "direction" as being vague and indefinite, Applicants respectfully submit that use of this word in the claims particularly points out and distinctly claims the subject matter which Applicants regard as the invention. Applicants refer the Examiner to the Webster's New Universal Unabridged Dictionary definition of "direction." Webster defines a direction as an "instruction for doing, operating, using, preparing, etc." Applicants submit that use of "direction" in the pending claims incorporates this definition.

Further, regarding the Examiner's suggestion that "the" be replaced by "said" in the claims, Applicants respectfully submit that these two words are well-recognized equivalents. Accordingly, Applicants wish to leave occurrences of "the" unamended. In view of the above, Applicants respectfully request that the Examiner reconsider and withdraw the rejection of Claims 3 and 13 under 35 U.S.C. § 112, second paragraph.

The Office Action rejected Claims 1-36 and 44-55 under 35 U.S.C. § 103 as being unpatentable over Harmon et al., "the Motorola MC68000," in view of Sebesta, "Concepts of Programming Languages." Applicants respectfully traverse.

Before setting forth Applicants' somewhat lengthy and involved arguments, Applicants wish to thank Examiner Kang for the several interviews extended on May 3 and May 4, 1995. In these interviews, Examiner Kang was very patient and cooperative in listening to Applicants' description of the subject invention and claims.

Since Examiner Kang requested that Applicants submit their arguments in writing, Applicants have done so below. Applicants have made efforts to be clear and concise, and respectfully request that the Examiner closely consider the below comments.

In the course of these several telephone interviews, Applicants faxed Examiner Kang several proposed amendments to Claims 1 and 13. Examiner Kang stated that these amendments were definitely steps in the right direction, but was unwilling to acknowledge that either of these two amended claims are allowable.

Examiner Kang stated, however, that the compensate instruction generating means of Claim 13 may be allowable over the prior art of record, since he was not readily aware of a reference teaching or suggesting the structure of this claim. In response to Examiner Kang's request, Applicants are reminding Examiner Kang of his observations that amended Claim 13 may be allowable.

The two amended Claims 1 and 13, previously submitted to Examiner Kang, are set forth in this Amendment as new Claims 56 and 57, respectively.

I.

Claims 1-12 and 56; Embedded Microprocessor Series, Etc.

Regarding new Claim 56 (which corresponds to amended Claim 1), the Office Action and Examiner Kang are basically relying on bit expansions performed by modern processors to accommodate programs written for older processors. Specifically, the Office Action states on page 4 that:

> The reason for bit expansion is so that new processors that works with wider word (64 bits) can be compatible with older processors. Also it makes the programmer's job easier by not having to specifically give full bit width every time they are instructing to perform calculations. For example programmer does not have to always type: ADD Hex.0001,R1 for 16 bit architecture, instead ADD Hex.1,R1 would do the same.

According to Examiner Kang, Claim 56 (and corresponding Claims 1 and 6) are rejected as being obvious in view of a 486 microprocessor which can run codes written for 286, 386, and 486 microprocessors. Applicants respectfully traverse.

The prior art relied upon is similar to the third conventional processor disclosed on pages 6-12 of Applicants' specification. Exhibit A (attached hereto) shows such a conventional program converter, which can convert source programs having different data and address bit widths to a single, predetermined machine language program with a single data and address bit width.

In contrast to this prior art architecture, the program converter recited in Claims 1 and 56 converts any source program to any of a plurality of machine language programs for use by a corresponding plurality of microprocessors in a custom microprocessor series. (See bottom of Exhibit A.) Each microprocessor in the custom microprocessor series has a different address bit width. Each microprocessor also has a different memory size bit width, program counter bit width, register bit width, and ALU bit width corresponding to the address bit width of that microprocessor. Looking at the bottom of Exhibit A, a source program having 16-bit data may be converted into machine language code for use by a microprocessor having a 16-bit address bit width, a 17-bit address bit width, an 18-bit address bit width, or a 24-bit address bit width. The address bit width of the output machine language program is determined by a user input. This input is called the pointer width N, and is selected by the user based on the size of the source program to be converted. (In Applicants' specification, a 24-bit address width (N) and a 16-bit data width (M) are disclosed as an illustrative example. Applicants' Claims 28-36 disclose a custom size microprocessor.)

The program converting unit of Claim 56 (and Claims 1-12) comprises a parameter holding means and a generation means. The parameter holding means holds the data bit width M and the pointer bit width N. The data bit width M represents the bit width of data in the source program to be converted, and the pointer bit width N represents the address bit width that the machine language program should have. This address bit width N enables the program converting unit to identify which particular microprocessor in the microprocessor series will be running the machine language program.

Before a source program can be converted to any of the plurality of machine language programs (for use by a corresponding one of the plurality of microprocessors), a user must determine a program capacity required by the source program in order to input a particular pointer bit width N to designate one of the microprocessors having a suitable address bit width to handle execution of that source program. The generation means of Claim 56 (and correspond Claims 1 and 6) generates the machine language instructions for the selected microprocessor in the custom microprocessor series.

In summary, a conventional program converter converts a source program into a single, predetermined bit width machine language program. No user input is required.

In contrast, the presently claimed invention does not convert a source program into a single, predetermined machine language program. The presently claimed invention converts the source program into any of a plurality of machine language programs, each having a different address bit width and corresponding to a different microprocessor.

Since the conventional program converter does not selectively convert the source program into any of the plurality of machine language programs, it does not accept a user input based on the size of the input source program. Neither Harmon et al. nor Sebesta disclose any user input based on the size (N) of the source program, and a corresponding number (N) of microprocessors having different address bit widths. The prior art relied upon only shows a program converter for converting a source program into a single type of machine language for a single microprocessor.

II.

Claims 13-27, 57; Option Detecting Means, Etc.

Regarding new Claim 57 (and Claims 13-27), Applicants respectfully submit that this independent Claim 57 recites the "compensate instruction generating means," "option direction means," and "prohibition means" that Examiner Kang indicated may comprise allowable material.

Conventional program converters generate compensate instructions for each machine language instruction that presents the possibility of an overflow. An overflow can occur, for example, when the addition of two 16-bit registers yields a 17-bit result.

The MC68000 represents a typical CISC microprocessor. When this microprocessor executes "ADD.W #1, D0" and the result is an overflow, the overflow situation is corrected by the D0 register holding 0 and a V-flag being set, thus allowing the execution result to be within the range of the short integer variable (16 bits, for example).

Conventional RISC compilers add a compensate instruction to clear all of the bits beyond the 17th bit to any machine language instruction which has a possibility of an overflow. Although this conventional compensate instruction eliminates the aforementioned overflow drawback, it must be executed for every single operation that presents a potential overflow. Accordingly, program size and execution time are increased.

Applicants' invention recognizes that this increase in program size and execution time can be avoided when a user determines that no overflow for a given operation will occur (even if an overflow possibility theoretically exists). New Claim 57 and Claims 13-27 all recite an option directing means for holding a user's direction as to whether a compensate instruction is necessary. This feature allows programming versatility, reduces program size, and improves program execution speed.

The prior art reference Harmon et al. does not even recognize the problems solved by Applicants' claimed invention. Specifically, Harmon et al. does not mention the undesirable expansion of program size and running time associated with generating a compensate instruction for machine language instructions. Accordingly, the option directing means and inhibit means of the presently claimed invention are neither disclosed nor suggested by the prior art of record.

III.

Claims 28-36; a Microprocessor of the Embedded Microprocessor Series, Etc.

Regarding the rejection of Claims 28-36, Applicants respectfully submit that there is no teaching in the prior art for an embedded custom microprocessor series comprising a plurality of microprocessors having different address bit widths. The embedded custom microprocessor series of the present invention provides versatility of function and allows a user to customize the entire device according to an intended application. Video tape recorders, television sets, production line control devices, and computer peripherals are examples of devices featuring such embedded-use microprocessors. Thus, when a user selects one of the microprocessors in the embedded microprocessor series, he or she is, in effect, selecting an address bit width processing for a specific microprocessor. The selected address bit processing protocol thus covers memory size ("memory means" in Claim 28), the number of bits for the program counter ("Program Counter" in Claim 28), the number of bits for the registers (especially the "address register group" in Claims 29 and 33), and the number of bits of the ALU ("execute means"). The embodiment illustrated in Applicants' specification discloses an illustrative situation where the address bit width N is set at 24 and the data bit width M is set at 16.

According to Applicants' invention, a variety of address bit widths N can be used, such as 17, 18, 19, 20, etc. The apparatus of the present invention is thus designed to allow all arithmetic operation instructions to be executed at a selected bit width N, thus allowing the execution means to effectively perform address calculations according to the specified bit width N. Thus, the embedded microprocessor series of the present invention can compile and execute a first simple program with 16-bit addressing, and subsequently execute a subsequent complex program with 24-bit addressing.

In contrast, conventional processors set the number of address bits to a single size that is large enough for applications of varying program size. When a number of applications are executed repeatedly, with the program size of such applications varying greatly, resources are not conserved. Additionally, the number of data bits (which is unrelated to the number of address bits) is set by the prior art as the most important aspect in determining the ability of the microprocessor. Conventional address calculation is thus performed by utilizing a variety of data-calculating instructions.

Accordingly, the prior art neither discloses nor suggests selecting a microprocessor based upon the size of the program to be executed and the address bit width of the microprocessor. The prior art neither discloses nor suggests Applicants' claimed memory means, having a capacity that is customized to the address bit width N input by a user according to program size, an N bit program counter, the execution means for executing all arithmetic operation instructions at N bit length, and a combination thereof. The prior art merely incorporates a single microprocessor with an address bit width large enough to accommodate all programs to be executed on the microprocessor.

IV.

Claims 37-43, Address Bit Width of Selected Microprocessor Corresponding to Estimated Object Program Size, Etc.

Regarding Claims 37-43, these claims recite a method of communicating the number of bits to be transmitted without using a size field. The transmission instruction code of the prior art incorporated a size field, denoting the number of bits of data to be transmitted. The MC68000 is an example of this architecture. The MC68000 architecture results in an increase in instruction code and program size, and a corresponding decrease in the number of instructions that can be assigned to the instruction code. This architecture further results in an increased complexity and size of the instruction decoding unit 12, which decodes the size field.

The instruction code of the present invention, on the other hand, incorporates an external access-width control means that conveys the number of bits to be transmitted in accordance with the kind of register used. Thus, the size field of the prior art is not needed, since the same information is conveyed through the selection of registers.

As disclosed in Applicants' specification, for example, 24-bit transmission is designated when the address register is used, and 16-bit transmission is designated when the data register is used. Claim 39 recites that the number of bits to be transmitted is set in accordance with the bit length of the address register and the bit length of the data register. Specifically, as recited in Claim 40, the address register sets the length to 24 bits, and the data register sets the length to 16 bits. According to Claim 37, it is unnecessary to specify the number of bits of data to be transmitted in the instruction code and, consequently, a larger number of instructions can be allocated to the instruction code. Moreover, the instruction decoding means is no longer required to be equipped with a function for decoding the size field. Neither Harmon et al. nor Sebesta, taken together or separately, disclose or suggest, among other things, a protocol where the number of bits to be transmitted is designated according to an external indication (the specific register used).

V.

Claims 44-50; Register Designation Determines Type of Immediate Data Extension, Etc.

Claims 44-50 distinguish over the prior art of record as well. The invention recited in these claims operates in situations where immediate data (which is shorter than the bit width of a storage register) is to be stored in the storage register. In such a situation, the immediate data must be extended to the bit width of the storage register.

In the prior art, the immediate data is extended with zeros or code, according to a demarcation in the instruction code. That is, the instruction code of the prior art contains information designating whether the immediate data should be zero or code extended. This prior art approach unnecessarily consumes instruction code. Other conventional microprocessors, such as the MC68000, extend immediate data with code, regardless of whether the immediate data is to be stored in an address register or a data register.

In either of these two prior art architectures, the instruction decoder, in decoding the instructions, differentiates between instructions with zero extension and those with code extension. Immediate data extended with only code by the MC68000 must subsequently be subjected to a mask process in order to obtain the same results for this immediate data as would be obtained using a zero extension. (See page 12, lines 2-8 of Applicants' specification.) Thus, one approach of the prior art unnecessarily consumes instruction code, and the other approach of the prior art requires a subsequent mask process.

Applicants' claimed instruction control means of amended Claim 44 (the "instruction decoding means" of Claim 47) determines whether to perform zero or code extension, based on the type of register in which the immediate data is to be stored.

Specifically, zero extension is performed when the immediate data is to be stored in a first register means, and code extension is performed when the immediate data is to be stored in a second register means. Thus, the information designating whether a zero or code extension should be performed is conveyed by the type of storage register designated for storing the immediate data. With the present invention, it is unnecessary to provide any explicit demarcation of whether zero or code extensions should be applied to the immediate data. Applicants respectfully submit that the prior art of record neither discloses nor suggests any similar architecture.

VI.

Claims 51-55; Method Claims and Claims Reciting Plurality of Groups of Flags, Etc.

Claims 51-53 are method claims relating to the process disclosed by Claims 44-50, and are allowable for at least the reasons discussed above with reference to Claims 44-50.

Claims 54 and 55 basically include all of the limitations of Claims 28, 37, and 44, and the construction elements relating to the flags (the plurality of flag storing means, the flag selecting means, and the branch judging means). These Claims 54 and 55 are allowable for at least the reasons that Claims 28, 37, and 44 are allowable.

Regarding the construction elements relating to the flags, Applicants respectfully submit that conventional microprocessors are only provided with a single group of flags. As an example, the MC68000 is only equipped with one group of five flags comprising X/N/Z/V/C flags. In contrast, Applicants' claimed invention, as recited in dependent Claim 54, recites a plurality of groups of flags.

According to conventional microprocessor architecture, the number of bits for an operation is specified in the size field of the instruction, and the operation result for the specified number of bits for the operation is shown in the single group of flags. For example, when a 16-bit data word is specified in the size field of an instruction, every flag in the single group of flags is set/reset based on the word data operation result. A flag may be set, for example, depending on whether there is a carry from bit 15 to bit 16. When a 32-bit data word is specified in the size field of the instruction, for example, every flag in the single group of flags is set/reset based on the long word data operation result. Here, an example may designate whether a carry has occurred from bit 31 to bit 32. From the above, it can be seen that only one group of flags is necessary for the prior art, so long as the size field is specified in the instruction.

In contrast, Claim 54 requires a plurality of flag storing means. The plurality of flag storing means are used to show operation results of different bit widths. Since the execution means of the present invention executes all arithmetic operations at N bit length (that is, the bit length the first and second register means is in), supposing that one flag storing means represents an operation result as 24 bits and another flag storing means represents an operation result as 16 bits, then one of the two storing means will be used to reflect the operation result depending on its bit length. For this method, using a number of groups of flags, the conditional judgment can be made on the comparative size of the bit length of the operation result, and the branch selecting means can select which flag storage means to use according to a conditional jump instruction. By doing so, it becomes unnecessary to provide a size field in the instruction and, accordingly, simplified instruction code may be used. The prior art neither discloses nor suggests Applicants' claimed combination, including a plurality of flag storing means, the flag selecting means, and the branch storing means.

In view of the above amendments and comments, Applicants respectfully request that the Examiner reconsider and withdraw the rejection of Claims 1-55 under 35 U.S.C. § 103.

It is respectfully submitted that the case is now in condition for allowance, and an early notification of the same is requested. If the Examiner believes that a telephone interview will help further the prosecution of this case, Applicants respectfully request that the undersigned attorney be contacted at the listed telephone number.

Respectfully submitted,

PRICE, GESS & UBELL

Joseph W. Price
Registration No. 25,124
2100 S.E. Main St., Suite 250
Irvine, California 92714
Telephone: 714/261-8433

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to the Commissioner of Patents and Trademarks, Washington DC 20231 on May 12, 1995

Quynh Vu

Quynh Vu
Signature

May 12, 1995
Date